

## Help Page

### RepWords 1.0 Program

---

RepWords 1.0 is intended for detecting repeats in sequences given in FASTA format with an arbitrary scoring matrix and affine gap penalties.

The program can be downloaded from the URL:

[http://www.ncbi.nlm.nih.gov/CBBresearch/Spooge/html\\_ncbi/html/index/software.html#18](http://www.ncbi.nlm.nih.gov/CBBresearch/Spooge/html_ncbi/html/index/software.html#18)

Instructions for the installation can be found [here](#).

### Usage.

---

The program is run with parameters separated by spaces. Each parameter of the command line is the pair:

-<parameter name> <parameter value>

Some parameters are optional but some are required; if a parameter cannot be used in the context of other parameters, it must be excluded from the command line (otherwise the program returns an error message). The order of the parameters is not important.

### Parameters.

---

The program can be executed in the three different modes:

- Mode A: calculation of repeats for a single  $w$ .
- Mode B: combining of the results for different  $w$ .

- Mode C: repeats calculation for different **w** with further combining of the results: Mode A + Mode B.

The opening gap penalty **d1** and the extending gap penalty **d2** (defined in Mode A and Mode C of the program) assume the following convention: a gap of length **k** is penalized as **d1+d2\*k**.

### ***Mode A (calculation of repeats for a single w)***

**-w <word-length w>**

- Must be a positive integer number.
- The parameter is required.

**-gap\_open <opening gap penalty>**

- Must be a non-negative integer number.
- The parameter is required in the case of the gapped repeats (when “**-gapped true**” is defined).

**-gap\_extend <extending gap penalty>**

- Must be a positive integer number.
- The parameter is required in the case of the gapped repeats (when “**-gapped true**” is defined).

**-gapped <gap penalty flag>**

- Defines whether the repeats are gapped or not; the setting “**-gapped true**” corresponds to gapped repeats.
- The parameters “**-gap\_open**” and “**-gap\_extend**” must be defined in the case “**-gapped true**”.
- The setting “**-gapped false**” corresponds to the gapless case and the parameters “**-gap\_open**” and “**-gap\_extend**” must be excluded from the command line in this case.
- The parameter is optional (the default value is “**-gapped true**”).

**-scoring\_matrix <a name of an input file with the scoring matrix>**

- The format of the file as follows. The first line contains a positive integer number **B** of letters in the alphabet. The rest of the file is a **BxB** table with **B** rows and **B** columns. The element from the row **a** and the column **b** of the table is an integer number corresponded to the similarity score between the letters with the order numbers **a** and **b**.
- The parameter is required.

**-alphabet\_yes <a name of an input file with a list of allowed letters>**

- The order of the letters corresponds to the order of columns and rows of the scoring matrix.
- The parameter is required.

**-alphabet\_no <a name of an input file with ignored letters>**

- Contains additional letters permitted in the input file; these letters are ignored.
- All ignored letters must be defined; otherwise, if one of non-listed letters will be encountered in the input file, the program will return an error.
- Any chain of the ignored letters breaks a repeat so a repeat cannot contain any of these letters.
- An alternative to the parameter “**-alphabet\_no**” is to include the ignored letters into the list defined by the parameter ”**-alphabet\_yes**” and to construct the scoring matrix in the way that the ignored letters do not match each other and do not match to the allowed letters.
- The parameter is optional (if it is not defined, then the set of the ignored letters is empty).

**-FASTA\_input <a name of an input file with sequences in FASTA format>**

- Each sequence may be presented as a single line or several lines.
- The parameter is required.

**-FASTA\_output <a name of an output file in FASTA format>**

- The parameter is required.

**-HTML\_output <a name of an output HTML file in FASTA format>**

- The parameter is optional (there will be no an HTML output if not defined).

**-HTML\_color\_factor <a factor for red color>**

- Defines a factor by which the program multiplies a repeat's score to produce a hue of red color for displaying the repeat in the HTML output file.
- The color belongs to [0,255]; the greater the brighter; if the color (the score multiplied by the factor) is greater than 255, then the color is replaced by 255.
- Valid for the HTML output only.
- The parameter is optional (the default value is 1000 what corresponds to uniform marking of repeats by red color).

**-output\_line\_size <length of lines of the output files>**

- Can be set to some large number like 1000000000 to ensure one-line output.
- The parameter is optional (the default value is 70).

**-show\_gaps <gaps flag>**

- Defines whether to show ("**-show\_gaps true**") or not ("**-show\_gaps false**") gaps (symbols '**-**') in the output.
- The parameter is optional (the default value is "**-show\_gaps false**").

**-score\_threshold <score threshold>**

- The program only marks repeats with this score or greater.
- The parameter is optional (the default value is -1 what corresponds to marking all repeats since a repeat's score is always greater than 0).

**-array\_seed\_length <an increment C for dimensions of internal arrays>**

- Total memory used by the program is guaranteed to be **a\*C+b** where **a** and **b** are some fixed values. Very small values are not recommended for long input sequences. Very large values like 10,000,000 or greater (depending on C++ compiler and the computer) can cause a memory allocation error and are not recommended.
- The parameter is optional (the default value is 100000).

## ***Mode B (combining of the results for different w)***

**-input\_w <a name of an input file with w-values>**

- Each line of the “-input\_w” file has the format:  
**w <a name of a text file precomputed in mode A>**
- The “-input\_w” file contains w-values and corresponded names of input text files; a text file listed for the value w contains an entire input sequence with w-repeats masked by low case letters. The text files can be precomputed in mode A (where a file name of the text output is defined by the parameter “-FASTA\_output”).
- The parameter is required.

**-alphabet\_yes <a name of an input file with a list of allowed letters>**

- The parameter has the same meaning as in Mode A.
- The parameter is required.

**-alphabet\_no <a name of an input file with ignored letters>**

- The parameter has the same meaning as in Mode A.
- The parameter is optional.

**-FASTA\_output <a name of a text output file with combined results>**

- The parameter is required.

**-prerepeats <a flag specifying whether or not to mask w-1 letters preceding a repeat>**

- A Boolean parameter that determines whether or not the program masks w-1 letters preceding a repeat (the program masks for “-prerepeats true” and does not for “-prerepeats false”).
- Mode A does not have functionality corresponded to this option.
- The parameter is optional (the default value if “-prerepeats false”)

**-HTML\_output <a name of an output HTML file with combined results>**

- The parameter is optional (the program does not produce an HTML output if the parameter is missing).

**-input\_colors <a name of an input file with colors>**

- The colors are used in the HTML output and the file explains how to color repeats corresponded to different **w**.
- Each line has the format:  
`<w> <RGB hexadecimal code of the color>`
- Can be set only if the parameter “-HTML\_output” is defined.
- The file must contain all **w** listed in the file defined by the parameter “-input\_w” but may contain colors for **w** that are not used.
- The parameter is optional (the program uses default colors for each **w** if the parameter is missing).

**-explain\_colors <a name of an output file with explanations of colors for different w>**

- The parameter is optional (the file is not outputted if the parameter is missing).

***Mode C: calculation of repeats for different w with further combining of the results: Mode A + Mode B.***

All parameters from mode A and mode B are permitted in mode C besides the parameter “-w” of mode A. In mode C, **w**-values and corresponding text output file names for mode A are extracted from the file defined by the parameter “-input\_w” (therefore, mode A will be run for each **w** individually).

---

### **Examples of command line for different modes.**

All files required for the examples are available in the download directory.

**Mode A.**

An example of the command line:

```
-w 7 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -  
alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -  
FASTA_input seq.in -FASTA_output output_seq.txt -HTML_output  
output_seq.html -HTML_color_factor 10 -output_line_size 70 -show_gaps true -  
score_threshold 20 -array_seed_length 100
```

Below are explanations of each parameter.

**-w 7**: the word-length used for the calculation is **w=7**.

**-gap\_open 5, -gap\_extend 2**: a gap of length **k** is penalized as **5+2\*k**.

**-gapped true**: the calculation is gapped.

**-scoring\_matrix matr4.in**: the scoring matrix is inputted from the file “**matr4.in**”.

**-alphabet\_yes alphabet\_ACGT.in**: allowed letters are inputted from the file “**alphabet\_ACGT.in**”.

**-alphabet\_no alphabet\_NYKWRSM.in**: ignored letters are inputted from the file “**alphabet\_NYKWRSM.in**”.

**-FASTA\_input seq.in**: an input file with sequences is “**seq.in**”.

**-FASTA\_output output\_seq.txt**: the results are outputted into the file “**output\_seq.txt**” in the text format.

**-HTML\_output output\_seq.html**: the results are outputted into the file “**output\_seq.html**” in the HTML format.

**-HTML\_color\_factor 10**: the multiplier for red color is 10.

**-output\_line\_size 70**: lines of the output files have length 70.

**-show\_gaps true**: the gaps are displayed as the symbol ‘-’.

**-score\_threshold 20**: the repeats are filtered by the score threshold 20.

**-array\_seed\_length 100**: the seed used for internal arrays is 100000.

### **Mode B.**

An example of the command line (to run this command, mode A must be executed for  $w=1, \dots, 10$  to produce output files with names as they are defined in “*w.in*”; please see the file ”*mode\_A.bat*” as an example of a batch file for Windows):

```
-input_w w.in -alphabet_yes alphabet_ACGT.in -alphabet_no  
alphabet_NYKWRSM.in -FASTA_output output_seq01_10.txt -prerepeats true -  
HTML_output output_seq01_10.html -input_colors colors01_10.in -explain_colors  
colors01_10.html
```

Below are explanations for each parameter.

**-input\_w w.in:**  $w$ -values are read from the file “*w.in*”.

**-alphabet\_yes alphabet\_ACGT.in:** allowed letters are inputted from the file “*alphabet\_ACGT.in*”.

**-alphabet\_no alphabet\_NYKWRSM.in:** ignored letters are inputted from the file “*alphabet\_NYKWRSM.in*”.

**-FASTA\_output output\_seq01\_10.txt:** the program outputs the combined result into the file “*output\_seq01\_10.txt*” in the text format.

**-prerepeats true:**  $w-1$  letters preceding a repeat are marked.

**-HTML\_output output\_seq01\_10.html:** the program outputs the combined result into the file “*output\_seq01\_10.html*” in the HTML format.

**-input\_colors colors01\_10.in:** the program inputs colors for different  $w$  from the file “*colors01\_10.in*”.

**-explain\_colors colors01\_10.html:** the program outputs information about each color into the file “*colors01\_10.html*” in the HTML format.

### **Mode C.**

An example of the command line:

```

-gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes
alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -
HTML_color_factor 10 -output_line_size 70 -show_gaps true -score_threshold 20
-array_seed_length 100 -input_w w.in -FASTA_output output_seq01_10.txt -
prerepeats true -HTML_output output_seq01_10.html -input_colors
colors01_10.in -explain_colors colors01_10.html

```

Explanations of the parameters are the same as for modes A and B.

### **Files required for the examples.**

---

The following files are provided in the folder “Examples”.

1) ”matr4.in”: an example of a scoring matrix for the alphabet “ACGT”:

4			
4	-5	-5	-5
-5	4	-5	-5
-5	-5	4	-5
-5	-5	-5	4

2) ”alphabet\_ACGT.in”: the alphabet “ACGT”:

4
ACGT

3) ”alphabet\_NYKWRSM.in”: the additional letters “NYKWRSM”:

7
NYKWRSM

4) ”matr4\_NYKWRSM.in”: a scoring matrix for the alphabet “ACGTNYKWRSM”=“ACGT”+“NYKWRSM”. Large penalties are assigned for the additional letters.

```

11
2   -3    -2    -3    -1000   -1000   -1000   -1000   -1000   -1000   -1000
-3    5    -3    -3    -1000   -1000   -1000   -1000   -1000   -1000   -1000
-2    -3    5    -2    -1000   -1000   -1000   -1000   -1000   -1000   -1000
-3    -3    -2    2    -1000   -1000   -1000   -1000   -1000   -1000   -1000
-1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000
-1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000
-1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000
-1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000
-1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000
-1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000  -1000

```

5) "colors01\_10.in": a possible choice of the colors. Different colors are assigned for  $w=1,\dots,10$ :

1	0xFF0000
2	0x00DD00
3	0x0000DD
4	0xFFD700
5	0x00DDDD
6	0x800000
7	0x008000
8	0x000080
9	0x008080
10	0xFF00FF

6.A) "mode\_A.bat" is a Windows batch file to run the program in mode A for each  $w$  from the interval [1,10]:

```

sls_repwords.exe -w 1 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w1.txt -HTML_output w1.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 2 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w2.txt -HTML_output w2.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 3 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w3.txt -HTML_output w3.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 4 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w4.txt -HTML_output w4.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 5 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w5.txt -HTML_output w5.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 6 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w6.txt -HTML_output w6.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 7 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w7.txt -HTML_output w7.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 8 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w8.txt -HTML_output w8.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 9 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w9.txt -HTML_output w9.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100
sls_repwords.exe -w 10 -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -FASTA_output w10.txt -HTML_output w10.html -
HTML_color_factor 1 -output_line_size 70 -show_gaps true -score_threshold 20 -array_seed_length 100

```

6.B)**"mode\_B.bat"**: an example of the command line of the program in mode B:

```

sls_repwords.exe -input_w w.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_output
output_seq01_10.txt -prerepeats true -HTML_output output_seq01_10.html -input_colors colors01_10.in -explain_colors
colors01_10.html -summary summary001_100.txt

```

6.C)**"mode\_C.bat"**: an example of a command line in mode C:

```

sls_repwords.exe -gap_open 5 -gap_extend 2 -gapped true -scoring_matrix matr4.in -alphabet_yes alphabet_ACGT.in -alphabet_no alphabet_NYKWRSM.in -FASTA_input seq.in -HTML_color_factor 1 -output_line_size 70 -show_gaps true -
score_threshold 20 -array_seed_length 100 -input_w w.in -FASTA_output output_seq01_10.txt -prerepeats true -
HTML_output output_seq01_10.html -input_colors colors01_10.in -explain_colors colors01_10.html -summary
summary01_10.txt

```

7)**"w.in"**: a sample file for the parameter "**-input\_w**":

1	w1.txt
2	w2.txt
3	w3.txt
4	w4.txt
5	w5.txt
6	w6.txt
7	w7.txt
8	w8.txt
9	w9.txt
10	w10.txt

8) "seq.in": a sample input FASTA file with a single sequence (a fragment of human chromosome 19).

9) "blosum62.in", "blosum62\_XUBJZ.in", "alphabet\_ARNDCQEGHILKMFPSTWYV.in", "alphabet\_XUBJZ.in": sample files to work with sequences containing amino acids.

Please copy these files into a folder with an executable to run the examples.

### **Additional comments.**

### *Mode A.*

The program calculates repeats for a single w defined by the parameter “-w”.

### *Mode B.*

If the calculation in mode A is repeated for different  $w$  (all other parameters must be identical), then we can combine the results into a single output in mode B. Only text (not HTML) files from mode A are used as an input in mode B. Mode B also requires files with permitted and ignored alphabet letters, and they must be identical to the ones used for the calculation in Mode A.

The rule of the combining is the following. If for a specific letter from an input sequence there exists  $w$  such that the letter belongs to a  $w$ -repeat, then the letter is marked as a repeat in the combined output. For the HTML output, the program finds the minimum  $w$  such that the letter belongs to  $w$ -repeat and assigns the color corresponded to this  $w$  to the letter in the combined output. The program outputs the symbol exactly as it is outputted for this minimum  $w$  in the input file from mode A (it can be a gap symbol ‘-’ or a low case letter). The program outputs the original upper case letter if it is not inside of a repeat for all  $w$  considered, but outputs the ignored letters as low case letters.

### ***Mode C.***

In mode C, the program applies mode A for different input  $w$ ; then applies mode B for the calculated results; and finally outputs the combined results. The command line in mode C contains parameters from both mode A and mode B except for the parameter “`-w`” of mode A. The  $w$ -values are extracted from a file defined by the parameter “`-input_w`” and this file also contains names of the text files outputted in mode A for different  $w$ . The program outputs these text files and if the parameter “`-HTML_output`” is defined, then the program also outputs HTML files generated in mode A (the HTML file names are formed by the addition the extension “`.html`” to names of corresponding text files (defined in “`-input_w`”-file)). The combined result (generated in mode B) is outputted into the files with the names defined by the parameters “`-FASTA_output`” (text output) and “`-HTML_output`” (HTML output).

---

### **Some observations.**

- (i) One important parameter is the word-length  $w$  corresponded to a period of letters inside a repeat.

Examples of exact repeats:

$w=1$ : "AAAAAAAAAAAAAAA"

$w=5$ : "ACCTAACCTAACCTAACCTA"

The program calculates repeats according to a scoring matrix so a typical repeat is not an exact repetition. A repeat can look like this:

$w=1$ : "AAAGAAAAACCCCCC"

$w=5$ : "ACGTAACCTAAACTAACCTA"

- (ii) The program outputs the results in the FASTA format in two different forms: text output and HTML output. The HTML output file can be viewed by any Internet browser and marks repeats by different colors depending on the user's choice. HTML output is not recommended for very long sequences (longer than 20,000,000) since an Internet browser is not able to open a large file. The program works with large sequences however and an appropriate text (and HTML if required) output is produced (a sequence with the length greater than 240,000,000 was successfully tested with the current version).

Both text and HTML outputs use lower case letters to indicate repeats.

- (iii) The repeats for different  $w$  can overlap. We can expect that a strong  $w=a$ -repeat is also a  $w=a^*n$ -repeat for small  $n=1,2,\dots$ . Since the program uses a formal definition of a repeat with gap penalties applying a filtering by scores to repeats, we may expect a complex behavior of repeats for different  $w$ .

---

### Frequently asked questions.

- 
- (i) What range of  $w$  is optimal?

Tests show that the optimal range of  $w$  for Simple and Low complexity repeats is about [1,20]; the optimal range of  $w$  for Satellite and SINE repeats is about [1,200].

(ii) What is the best choice of the score threshold?

The Auxiliary RepWords program 1.0 (available in the download directory) computes the score threshold for a given user-defined coverage. The coverage has a meaning of a fraction of repeats in a random sequence calculated by RepWords program with specified parameters.

(iii) What is the optimal choice of the gap penalties and the scoring matrix?

It can be decided by a visual inspection of the HTML output with “**-gapped**” option turned on. If there are a lot of gaps ‘-’, then it might indicate that the penalties are too weak. If there are too many repeats, it might indicate an incorrect choice of the scoring matrix. Generally a reasonable scoring scheme always detects strong repeats. But if a problem considered is sensitive to the presence of weak repeats, then it might be necessary to test more alternative scoring schemes.

## Files and Installation

---

The files in the download directory include:

1. **repwords\_1.0\_WINDOWS.zip**: Windows executable.
2. **repwords\_LINUX\_1.0.zip**: LINUX executable.
3. **repwords\_cpp\_files.zip**: C++ source files.
4. **repwords\_examples.zip**: examples (please see section “[Files required for the examples](#)” for more information).

- No special installation is required.
- The executable files can be downloaded, unzipped and run with the appropriate command line.
- Alternatively, the source C++ files can be downloaded, unzipped, and complied in a suitable C++ environment.

### **Remark.**

---

To compile the C++ files under UNIX, please replace the line

#define \_MSDOS\_

by the line

//#define \_MSDOS\_

in the file “[sls\\_repwords.h](#)”.